Refine Search

Search Results -

Terms	Documents
L7 and Internet	3

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L8

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Tuesday, March 01, 2005 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; THES=ASSIGNEE; PLUR=YES; OP=OR			
L8	L7 and Internet	3	L8
L7	L6 and (track$ or navigat$)	3	L7
L6	L5 or l4 or l3	3	L6
L5	6453356.pn.	1	L5
L4	6125352.pn.	1	L4
L3	6029165.pn.	1	L3
L2	L1 and (track$ or navigat$)	0	L2
L1	6175830.pn.	1	L1

END OF SEARCH HISTORY

27

L7: Entry 3 of 3 File: USPT Feb 22, 2000

DOCUMENT-IDENTIFIER: US 6029165 A
**** See image for Certificate of Correction ****
TITLE: Search and retrieval information system and method

Detailed Description Text (14):
The user terminal shown in FIG. 2 also includes a Random Access Memory in (RAM) 19, Read Only Memory (ROM) 18, and an I/O adapter 21 for connecting peripheral devices such as disk storage units 23 to the bus 16. A user interface adapter 25 for connecting several input devices is also included. Examples of possible input devices connected to the user interface adapter 25 include a keyboard 35, a mouse 29, a speaker 28, a microphone 33, and/or other user interface devices such as a touch screen or voice interface (not shown). A communication adapter 37 is included for connecting the user terminal to a communication network link 39. A graphical user interface 41 is also connected to the system bus 16 and provides the connection to a display device 43. It will be apparent to those in the art that the mouse 29 may be a typical mouse as known in the industry, a trackball, light pen, or the like.

Detailed Description Text (15):
The user terminal typically has resident thereon an operating system such as Windows.RTM., Windows NT.RTM., Apple System 7.RTM., IBM OS/2.RTM., or UNIX.RTM. software. The network also has a resident operating system, for example, Novell.RTM. Netware or Novell.RTM. Intranetware, among other possibilities. In the preferred environment, the desktop typically has Internet browser software, such as MS Internet Explorer or Netscape Navigator. In the alternative, the network software operating system may not be available separate from the work station operating system, and the network operating system may have an integrated Internet browser. Other alternatives for client and server software include Oracle.RTM. or Microsoft Sequel Server.

Detailed Description Text (32):
FIG. 7 shows how a user typically retrieves the electronic objects found by the system and personalize the target profile. When accessing the information system for the first time 10, the user may be presented with a list of high-level topic areas at step 112 from the library of topics 36 that is designed for the community. Pointers and metadata 117 can be configured to store crucial information about content sources such as source location, source type, and other specifications. The user can select topic areas of interest for which electronic objects may be retrieved at step 114. The user can then select at step 116 the electronic objects of interest. The system typically delivers the first few sentences of each electronic object to the user's viewing frame. The user then selects an electronic object to read at step 118. That target object may then be streamed in its entirety to the user in step 120 so that the user can view the complete document. If the user would like to view another document at step 122, she may simply select another object from the list of electronic objects. When the user is finished viewing the found objects, another topic can be selected from the list of topic areas 124. At step 126, the user can personalize the target profile or exit the system at step 128. As discussed in greater detail in relation to FIG. 12, the system may be

configured to keep track of which objects the user and all users choose to view in order to obtain statistical information about the popularity of objects. The continued steps of the user while personalizing the target profile and retrieving electronic objects are shown in FIG. 8.

Previous Doc Next Doc Go to Doc#

L7: Entry 1 of 3 File: USPT Sep 17, 2002

DOCUMENT-IDENTIFIER: US 6453356 B1
TITLE: Data exchange system and method

Brief Summary Text (15):
Process monitoring, tracing, and logging are provided to track the progress of data passing through the data exchange engine and to detect and correct processing errors. In the case of a processing anomaly, the data exchange engine effects a rollback of failed transactions to preclude the loss of data. Performance statistics may also be provided.

Detailed Description Text (39):
Also shown in FIG. 9 is a statistics monitor module 264 and an associated statistics log 276 which are used to provide monitoring and tracking of data as it moves through the data exchange system. The statistics monitor module 264 also provides historical performance information on queues and historical information on system resource usage. As will be described in greater detail hereinbelow, the statistics monitor module 264 provides a means for logging and tracing a given application. Logging reveals the state of the application at the time of an error, while tracing provides a description of all software events as they occur. The tracing information may be used for tracking the application, state, and other related operations. The tracing information may be used in conjunction with the logging information to determine the cause for an error since it provides information about the sequence of events prior to an error.

Detailed Description Text (118):
The logging and tracing utility provides for logging and tracing during execution of a component. As discussed previously, logging reveals the state of the component at the time of an error, while tracing provides a description of all software events as they occur. The tracing information may be used for tracking the component-state, and other related operations. It may be used in conjunction with the logging information to determine the cause of an error, as it provides information about the sequence of events prior to an error.

CLAIMS:

7. The method of claim 1, further comprising tracking each of the data streams during converting, identifying, or transforming operations.

17. The method of claim 10, further comprising tracking the data during converting, identifying, or transforming operations.

25. The method of claim 19, further comprising: tracking processing of the information having the generic format; and logging errors occurring during the processing of the information having the generic format.

49. The medium of claim 44, further comprising tracking the data during converting, identifying, or transforming operations.

56. The system of claim 51, further comprising means for tracking the data during converting, identifying, or transforming operations.

Previous Doc Next Doc Go to Doc#

L7: Entry 2 of 3 File: USPT Sep 26, 2000

DOCUMENT-IDENTIFIER: US 6125352 A
**** See image for Certificate of Correction ****
TITLE: System and method for conducting commerce over a distributed network

Abstract Text (1):
A system and method for conducting commerce over a distributed network manage merchant and product information in an electronic shopping basket, payment source information in an electronic wallet, and shipping address information in an electronic address book, all of such information being stored on a consumer computer. A commerce client running on the consumer computer is configured as a MIME handler and extends the functionality of a standard Web browser to support computer-based shopping. A merchant site Web server provides HTML-coded Web documents which describe merchant products and which host computer-based shopping options. The HTML-coded Web documents contain function-calling information by which consumer-selected options invoke shopping-related functions on either the merchant (server) computer or the consumer (client) computer. A consumer selects the options from within the Web browser to initiate shopping-related operations such as: retrieve product information from merchants on the World Wide Web, selectively store product information locally on the consumer computer, locally compare product information from different merchants, locally store payment source and shipping address information and selectively forward such information to merchant sites, order products from Web-based merchants, track the status of purchase orders, and receive instructional information on application usage.

Brief Summary Text (8):
Also, because existing computer-based shopping systems typically host most or all transaction options on the server side, the shopping experience often differs from merchant site to merchant site. Specifically, as the consumer moves from one merchant site to the next, the options presented to the consumer and the steps required to navigate and effectively use those options often differ significantly. Thus, to browse product information and purchase products and/or services offered by multiple merchants, consumers often must adapt to and learn the options offered by each merchant site.

Detailed Description Text (43):
The consumer computer 102 comprises a processing unit 110, a data storage area 112, as well as a monitor 114, keyboard 116, and a mouse 118. A standard Web browser 120 (such as, for example, Microsoft Internet Explorer 2.0 or Netscape Navigator 2.0) and a specialized commerce client process 122 execute on the processing unit 110.

Detailed Description Text (47):
A consumer uses the consumer computer 102 to shop for products such as, for example, an audio compact disc, a downloadable software program, a rare con, or a refrigerator, offered by merchants via the World Wide Web 108. Using the example of shopping for a refrigerator, the consumer uses the consumer computer 102 to establish a connection to the Internet and uses a Web browser to navigate World Wide Web 108 sites. Connecting to the Internet and browsing the WWW is well known and the steps involved will not be further described.

Detailed Description Text (50):

Generally, consumer-selectable options ("hypertext links") are also presented within the HTML document which, if selected by a consumer using the mouse 118 or the keyboard 116, cause the consumer computer 102 to transmit requests to the Web server 128 to retrieve and transmit additional HTML documents providing related or more detailed information. The consumer navigates additional hypertext links and browses additional HTML documents summarizing features of refrigerators sold by merchant site A.

Detailed Description Text (52):
Two days later, the consumer, again having time to shop, directs the consumer computer 102 to connect to merchant site B via the World Wide Web. Merchant site B also offers refrigerators for sale. After navigating through HTML documents describing the features of the refrigerators sold by merchant site B, the consumer selects a refrigerator, sets the refrigerator attributes as desired, and invokes the "Add Item to Shopping Basket" option. The commands to accomplish this are the same as those required to add the information about the merchant site A refrigerator to the shopping basket. The commerce client 122 responds to the invoked option by adding information about the merchant site B refrigerator to the gathered products database 148. The consumer then terminates the shopping session with merchant site B 106.

Detailed Description Text (93):
FIG. 4 illustrates the software architecture of the consumer computer 102. The architecture comprises a conventional Web browser 120 (such as Microsoft's Internet Explorer 2.0, or Netscape Navigator 2.0), the commerce client 122, and a commerce client object library 402, and a storage area 112. The commerce client object library 402 comprises a shopping basket object 404 having associated shopping basket methods 406, a wallet object 408 having associated wallet methods 410, and an address book object 412 having associated address book methods 414. The storage area 112 comprises merchant information 416, product information 418, payment source data 140, and shipping address data 142, all of which has been selectively stored and/or manually entered by the user.

Detailed Description Text (108):
The commerce client 122 uses function-calling information embedded in the passed MIME message to call a method AddLineItem. In the step 612, the AddLineItem method navigates memory structures (constructed in step 608) looking for a merchant data structure with a name field matching the merchant name passed with the MIME message. If no such merchant structure is found in the step 612, then a new merchant structure is appended to the linked list of merchants by allocating memory and the new merchant structure is populated with merchant data from the passed MIME message. The AddLineItem method then, in a next step 614, navigates a linked list of product data structures associated with the merchant structure (either found or created in the step 612). Each product data structure in the list represents one product offered for sale by a merchant. In the step 616, the AddLineItem method allocates memory for a new product data structure and populates it with data packaged in the MIME message such as product SKU, price, quantity, description, name, logo, color, or size. The AddLineItem method then links the new product data structure to the end of the linked product data structures for the merchant.

Detailed Description Text (109):
In a step 618, the commerce client 122 invokes a SetItemProperty method wherein the linked list of merchant structures is again navigated until a merchant structure having a merchant name matching the passed merchant name is found. In a step 620, the product data structures referenced by the merchant structure are navigated until a product data structure is found having an SKU field equivalent to the SKU number passed in the MIME message. Then, in the step 622, the SetItemProperty method navigates a linked list of properties referenced by the product data structure found, and does so for each property passed in the MIME message. When a property structure is found whose name matches the passed property name, the value

associated with the matched property name is replaced with a value passed in connection with the property name. In each case where the property list is navigated and no matching property name is found, the SetItemProperty method appends a new property data structure to the linked list of properties. The SetItemProperty method allocates memory for the new property data structure and fills in the name and value and flag fields as appropriate from the data in the passed MIME message.

Detailed Description Text (113):
In a step 714, the commerce client 122 invokes a GetFirstItem method. In the step 714, the GetFirstItem method navigates to a given merchant. In this first iteration of the loop, the given merchant will simply be the first merchant in the merchant data structure list. In the step 716, the linked list of product data structures referenced from the first merchant data structure is navigated to the first product data structure. If, in the step 718, it is determined that there are no product data structures for this merchant or the navigation of the product data structure list has reached the end of the list, then, in a step 720, the GetFirstItem method returns a value of 0. If in the step 718 it is determined that another product data structure exists and that the end of the product data structure list has not yet been reached, then the flag field of the product data structure is compared against the CheckFlag variable. Because CheckFlag is equal to 0, the product flag is being compared against the value 0 which would indicate that the product has not yet been purchased. If in the step 722, it is determined that the product flag is not equal to the CheckFlag then processing reverts back to the step 716 to check the next product data structure in the list. If, however, in the step 722, the product flag is equal to the CheckFlag, indicating that the product has not yet been purchased, then in a step 724, a pointer is returned pointing to the product data structure of the product that has not been purchased and the name of the product that has not been purchased is added to a list box structure. Next, in a step 726 the pointer to the product data structure is saved and iterations of the loop are terminated.

Detailed Description Text (114):
In the step 728, the GetNextItem method is invoked which begins by navigating to a given merchant. In the step 730, a current chain of linked product data structures is navigated to a point equal to a saved pointer location (the pointer location saved in step 726 if this invocation of GetNextItem immediately follows the termination of the GetFirstItem method, otherwise the pointer location saved in the step 742 is used). Next, in the step 732, the next product data structure in the linked list is examined. If in the step 734 it is determined that the end of the product data structure list has been reached then in the step 736, a 0 is returned. If, however, in the step 734 it is determined that the end of the product data structure list has not yet been reached then, in the step 738, the product flag is compared against the CheckFlag. If in the step 738 the product flag is not equal to the CheckFlag then processing resumes in the step 732 examining the next product data structure in the list. If however, in the step 738 the product flat is equal to the CheckFlag then, in the step 740, a pointer is returned to the product data structure and the name of the product is added to the list box structure. Next, in the step 742, a pointer to the product data structure is saved. The GetNextItem method is then invoked again beginning at step 728 where the list of merchant data structures is navigated to the current merchant and the steps 728 through 742 are repeated until the end of the merchant data structure is reached.

Detailed Description Text (117):
In another embodiment, selecting the VIEW HISTORY option causes a list of all product orders to be displayed. Each merchant structure has a reference pointer to a list of order structures. Each order structure, in addition to having fields for payment, shipping, and order tracking information, also has a reference pointer to a list of product structures. Thus, the linked list of merchants is traversed. For each merchant, the respective merchant's list of order structures is traversed. One display item is generated from each order structure. Each display item shows on the

screen of the user computer, the purchase date, the payment information, the shipping address, the order tracking identifier, and the products ordered. Note, the products ordered are determined by traversing the linked list of product structures associated with each order structure. The VIEW HISTORY option advantageously allows consumers to examine past purchases whether such purchases occurred very recently or many years ago.

Detailed Description Text (129):
Next, a method GetNextProperty 924 is repeatedly called in a step 926 to navigate the linked list of property data structures. In a step 928, a list box is created having an entry for each property name/property value pair encountered in navigating the linked list of property data structures.

Detailed Description Text (130):
In a step 930, a GetPaymentFirstFriendlyName method 932 is invoked to examine the root pointer to the linked payment source structures and to return the value of the Friendly Name of the first payment source structure. A Friendly Name, as discussed below, is simply a name like "Bob's Visa Card" which is conveniently used to designate a payment source. In the step 934, the GetPaymentNextFriendlyName method 936 is used to navigate the linked list of payment source structures, placing the Friendly Name of each in the drop-down box already created in the step 916.

Detailed Description Text (131):
In a next step 938, a GetAddressFirstFriendlyName method 940 is invoked to examine the root pointer to the linked shipping address structures and to return the value of the Friendly Name of the first shipping address structure. A Friendly Name, as used with respect to an address, is a name like "the office" or "Debbie's house" which is conveniently used to designate a shipping address. In the step 942, the GetAddressNextFriendlyName method 944 is used to navigate the linked list of shipping address structures, placing the Friendly Name of each in the drop-down box already created in the step 918.

Detailed Description Text (156):
After transmitting an order, the commerce client 122 then updates the in memory structure for the respective merchant. Each merchant structure includes a reference pointer to a linked list of order structures. The commerce client 122 traverses the linked list of order structures, if any, to reach the final order structure. The commerce client 122 allocates memory for a new order structure, links the new structure to the list of order structures for the merchant, and then populates the new order structure with payment information, shipping address information, and a pointer to the product structure associated with the product ordered. An order tracking identifier ("OTI") field of the new order structure is left blank. One skilled in the art will appreciate that linked lists of structures is merely one way of associating data items together, and that other methods such as relational databases can be used to accomplish a similar association.

Detailed Description Text (157):
A merchant site receiving a product purchase order transmits an order confirmation message to a Web browser 120. The order confirmation message transmits an order tracking identifier ("OTI") to the Web browser 120. The Web browser 120 displays the OTI on the screen of the user computer, along with a message which states, for example, "Thank you for your order. If there is any problem, please phone 1 800 123 4567 and be prepared to refer to the following order tracking identifier." The OTI is also embedded in a MIME message of type x-ishopper which is passed by the Web browser 120 to the commerce client 122. The commerce client 122 then copies the OTI to the blank OTI field of the order structure, thus completing the storage of information related to the purchase.

Detailed Description Text (159):
An ORDER STATUS option is presented to a user by the commerce client 122 after

selection of the VIEW HISTORY option. Once a list of orders is generated and displayed to the user as described above, the user selects an order (preferably by positioning a mouse pointer over information associated with an order and clicking a mouse button). The user then selects the ORDER STATUS option. The commerce client then generates an HTTP POST message containing the order tracking identifier ("OTI"), and transmits the HTTP POST message to the order URL (uniquely identifying the Internet address of the selling merchant) associated with the product (or products) purchased. A merchant site receiving an HTTP POST message which includes an OTI determines the status of the order (e.g., "SHIPPED," "CANCELED," "WAITING FOR INVENTORY," etc.). The merchant site transmits an HTML document to the Web browser which includes the status information. The Web browser of the user computer displays the status information on the screen of the user computer.

CLAIMS:

25. A method as recited in claim 24, further comprising the step of selectively tracking status of the first ordering information and selectively tracking status of the second ordering information.

Previous Doc Next Doc Go to Doc#